



## Persistiendo una entidad #2

### Transcripción

[00:00] Hola. En el video anterior, nosotros creamos nuestra clase main donde instanciamos un primer producto, configuramos los valores para sus atributos e instanciamos el EntityManager, que es el encargado de realizar todas las operaciones de select, de update, de actualizar o de remover uno o más valores dentro de la tabla.

[00:21] Entonces nosotros ejecutamos esa aplicación y vimos que efectivamente nos mostró un registro en la consola. Luego de que nosotros obtuvimos ese registro nosotros aún no sabemos si el valor fue agregado en la base de datos. Entonces, para eso nosotros vamos a utilizar otra propiedad de Hibernate que nos va a permitir la operación que se está realizando en la base de datos.

[00:48] En el archivo de persistencia.xml, voy a agregar una nueva etiqueta de propiedades. Esa propiedad va a ser de hibernate y va a llamarse "hibernate.show\_sql", value="true", indicando que nosotros queremos que esta propiedad está activa.

[01:10] Vamos a ejecutar esta aplicación de nuevo y vemos que no se realizó ninguna operación en la base de datos. Eso es porque el EntityManager fue instanciado, nosotros ahora tenemos un valor para nuestro gerenciador, pero nosotros no le hemos indicado que debe comenzar las transacciones.

[01:30] Por eso nosotros vamos a llamar el EntityManager, vamos a obtener las transacciones y vamos a indicarle que las operaciones van a comenzar con el

método begin.

[01:42] Ya con esto, nosotros realizamos nuestra persistencia y por último vamos a realizar un commit. Vamos a obtener la transacción, `getTransaction()` y vamos a realizar un commit. Entonces, ese commit lo que hace es enviar los valores que fueron configurados para esta instancia los envía para la base datos.

[02:13] Por último vamos a cerrar y ahora vamos a ejecutar esta aplicación para ver si ahora sí está funcionando adecuadamente. Ahí genera el registro y genera un error. El error dice que la tabla de productos no fue encontrada.

[02:34] Entonces dice "la base de datos se encuentra vacía". Eso es porque nosotros hicimos las configuraciones de la base de datos, pero nosotros no creamos esa base de datos- Entonces nosotros vamos a utilizar otra propiedad de Hibernate, que va a construir esa base de datos y todas las columnas pertenecientes a ella.

[02:53] Vamos a crear el id, el nombre, la descripción y el precio. Entonces vamos al archivo de persistencia.xml, vamos agregar una nueva propiedad y la propiedad va a ser de `hibernate.hbm2ddl.auto`. Nosotros tenemos algunos valores para esta propiedad. El primer valor es create.

[03:22] Este valor indica que va a crear la base de datos, que va a crear las tablas, las columnas, va a insertar los valores y después de que haya finalizado la aplicación él va a dejar esos valores en la base de datos. Ellos van a permanecer ahí diferentemente que la aplicación esté corriendo o no.

[03:40] Vamos a tener create-drop, él va a crear la base de datos, va a agregar los valores y una vez que haya finalizado la aplicación va a eliminar todos los valores tanto de las tablas como de la base de datos. Vamos a tener otro valor que es validate, que no crea ni actualiza valores.

[04:04] Él simplemente verifica que los valores sean correctos, que existan esos valores, pero no hace, no agrega nuevos valores dentro de la base de datos, solamente para consulta. Y tenemos “update”, que él crea la tabla en caso de que no exista y en caso de que exista él realiza las operaciones y si está faltando un nuevo elemento, él lo crea.

[04:27] Luego de que finaliza la aplicación, estos valores permanecen dentro de la base de datos. Ese va a ser el valor que nosotros vamos a dejar para la construcción, vamos a ejecutar esta aplicación nuevamente y vemos que efectivamente él creó la conexión, conexión lo tenía de JDBC, eso fue cuando nosotros iniciamos la transacción, él creó la tabla producto con el id, la descripción, el nombre, el precio y realizó el insert cuando nosotros hicimos el commit.

[05:08] Eso fue nuestro primer intento de persistencia. Más adelante vamos a tener esto un poco más elegante, vamos a organizar este código un poco mejor, un poco más limpio y vamos a ver diferentes modelos de transacción.

[05:22] Hasta ahora, nosotros simplemente vimos cómo guardar un elemento, pero más adelante vamos a ver cómo actualizar ese elemento, cómo removerlo y cómo realizar consultas dentro de la base de datos.